MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ARTIFICIAL INTELLIGENCE LABORATORY

Artificial Intelligence Memo
No. 216

May 1971

THEORIES, PRETHEORIES,

AND FINITE STATE TRANSFORMATIONS ON TREES

Mitchell Wand

ABSTRACT: The closure of an algebra is defined as a generalization of the semigroup of a finite automaton. Pretheories are defined as a subclass of the closed algebras, and the relationship between pretheories and the algebraic theories of Lawvere [1963] is explored. Finally, pretheories are applied to the characterization problem of finite state transformations on trees, solving an open problem of Thatcher [1969].

Work reported herein was supported in part by the Artificial Intelligence Laboratory, an M.I.T. research program sponsored by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract number N00014-70-A-0362-0002, and in part by the author's National Science Foundation Graduate Fellowship.

0. Extended Abstract And Summary

Thatcher [1969] poses as an open problem the abstract definition of a 'pretheory'. A 'pretheory' is to be an algebraic structure which is to be intimately related (in some unspecified way) to the algebraic theories or algebraic categories of Lawvere [1963], and which will give some insight into the categorical approach to generalized finite automata via more familiar algebraic arguments. In addition, we are asked to define two pretheories such that every homomorphism from one to the other (again with an appropriate definition of homomorphism) corresponds to a finite-state transformation on a set of trees.

In this paper we propose a definition for a pretheory. Pretheories turn out to be algebraic structures called 'clones'. Taking the clone of an algebra is seen to be analogous to taking the semigroup of a finite-state automaton.

We explore the connection between theories and pretheories. We construct the free clone over a theory (in a manner similar to Eilenberg + Wright [1967]) and the free theory over an arbitrary set of maps (a construction alternative to that of Eilenberg + Wright). In the process of this exploration we introduce an intermediate structure, called a 'complete algebra,' which provides considerable insight into the definition of an algebraic theory.

Lastly, we apply pretheories to the characterization problem of finite-state transformations on trees, and complete the last portion of Thatcher's program.

In this section we intend to summarize our results using a minimum of symbolism, so as to explain the results rather than obscuring them by attempting a precise statement. We divide this summary into subsections, corresponding to the sections of the main part of the paper.

I. Algebras, Clones, and Pretheories

Following the usual terminology, let any set of function symbols with unique arities be called a 'ranked alphabet'. If V is a ranked alphabet, let a V-algebra be any interpretation of V (ie a model for the language). We call the domain of the interpretation the 'carrier' of the algebra.

If V is some ranked alphabet (either interpreted or uninterpreted) we will often use the symbol Vn to denote the set of n-ary operations in V. By a 'set of operations' we will mean

some interpreted ranked alphabet.

If V is a set of operations, we say it is a 'clone' [Gratzer 1968] iff

1) It contains all the projection functions, i.e. the function

lambda(x1,..,xn)[xi]

is in Vn for each i and n. We denote this function e(i;n) or just e(i).

2) It is closed under composition, i.e.: Let g be in Vn and f1,..,fn be in Vk. Let X denote the string 'x1,..,xk'. Then the function

lambda(X)[g(f1(X),..,fn(X))]

is in Vk. We denote this composite function by g(f1,..,fn).

This notation allows us to 'naturally' extend every operation in V to operate on arguments in V. Furthermore, if f1,..,fn are in Vk, then so is g(f1,..,fn). So for any k we have a V-algebra (Vk, V).

Definition. An algebra (V0, V), where V is a clone, is called a pretheory.

Note that every clone uniquely defines a pretheory, and vice-versa, so we will often refer to them interchangeably.

If V is any set of operations, define Cl(V), the clone of V to be the smallest clone containing V. If A is an algebra, let Cl(A), the closure of A, be the algebra whose carrier is the carrier of A and whose operations are the clone of the operations of A. Note that the closure of an algebra is a pretheory iff every element of the carrier is algebraic in the original algebra (i.e. the algebra has no proper subalgebras).

Theorem. Let V be any set of maps on a domain A. Then Cl(V) is just the set of maps A^* --> A achievable by composition of maps in V.

This shows that the closure of an algebra is analogous to the semigroup of a finite state machine, which is just the set of all maps from the state set to itself achievable by concatenation of symbols in the input alphabet.

Definition [Cohn]. Let V,W be clones. A Cl-morphism (clone homomorphism) is a map F:V --> W such that

1) $F(Vn) \subset Wn$

2) F(e(i;n)) = e(i;n)

3) F(g(f1,..,fn)) = (F(g))(F(f1),..,F(fn))

Proposition. Cl-morphisms compose.

2. Theories, pretheories, and complete algebras.

In this section we explore the relationship between Lawvere's notion of an algebraic theory and our notion of pretheory. In order to make Lawvere's ideas clearer, we will introduce an intermediate notion, which we will call a 'complete algebra.'

A complete algebra is not an algebra; it is the generalization of 'algebra' to functions into tuples of elements of A. The name 'complete' algebra was chosen because it is this type of structure which Arbib and Giveon [1967] call the completion of an algebra.

Definition. A complete algebra consists of a carrier A, and for each pair of non-negative integers m,n a set A(m,n) of maps A^m --> A^n such that

1) If $f \in A(m,n)$ and $g \in A(n,p)$, their composition, g.f, is in A(m,p)

2) The maps are closed under direct product, i.e. if $f1,..,fn \in A(m,1)$, there is a unique $g \in A(m,n)$ such that (letting X denote an m-tuple of elements of A)

g(X) = (f1(X),..,fn(X))

Note that the value of of g(a1,..,am) is indeed an n-tuple of elements of A. We write <f1,..,fn> for this unique g.

3) All the trivial maps are present; i.e. any direct product of the projections is present. (Note that with condition 2 it would have sufficed to require the inclusion of just the projections in A(n,1). The present definition is adopted in order to conform to the standard definition of the completion of an algebra.)

We prove the following characterization theorem for complete algebras: The mappings A(n,1) of a complete algebra form a clone. Furthermore a complete algebra consists precisely of all the direct products (axiom 2) of its maps A(n,1). Thus we can construct the completion of an algebra in two steps: First, take the closure of the algebra, and then take direct products.

We may then form the free theory of a complete algebra by letting T(m,n) be just A(n,m). This construction provides an alterate to that of Eilenberg and Wright [1967]. The proof that the so-called 'free theory' is indeed an algebraic theory, while trivial, provides insight into the definition of algebraic theory. It is seen that T is a category essentially because complete algebras are closed under composition; that S0 is a subcategory just because the trivial maps are in the algebra, and that the direct product condition for theories follows from the direct product condition for complete algebras. In this manner we obtain a deeper understanding of algebraic theories.

We can then construct the free theory over a pretheory by simply taking the free theory over its completion. We can also go in the other direction, constructing the free clone and free complete algebra over a theory. These constructions are seen to commute.

Furthermore, the notions of homomorphism of theory, complete algebra, and clone are shown to be related such that the following ladder diagram commutes:

$$\begin{tikzcd}
T1 \arrow[r] \arrow[d, leftrightarrow] & T2 \arrow[d, leftrightarrow] \\
CA1 \arrow[r] \arrow[d, leftrightarrow] & CA2 \arrow[d, leftrightarrow] \\
Cl1 \arrow[r] & Cl2
\end{tikzcd}$$

where the horizontal arrows are homomorphisms of the appropriate type, and the vertical arrows are any of the free constructions. In fact, given any of the horizontal maps, one may construct the other two homomorphisms such that the diagram commutes (no matter how the vertical arrows are drawn), and all these constructions themselves commute.

Thus the clones (and pretheories) are seen to be very intimately related to the algebraic theories of Lawvere.

3. Pretheories and Finite-State Transformations.

In this section we apply pretheories to the study of finite-state transformations on trees. This investigation was motivated by a remark of Thatcher [1969]. He noted that every finite state transformation on trees induced a homomorphism on a certain semigroup. He called these semigroups pretheories. Unfortunately, not every homomorphism on these semigroups was generated by a finite-state transformation. Thatcher conjectured that with the proper definition of pretheory, the converse of his observation would hold. We finish the program by presenting two pretheories such that the Cl-morphisms from one to the other correspond in a very natural way (in fact, by an elementary lambda conversion) to precisely the finite state transformations on the appropriate set of trees.

Let us use the symbol ':' to denote the center-dot substitution operator of thatcher (1969). If Σ is any set, let Trees(Σ; V) denote the set of all trees with nodes in the ranked alpphabet V and with additional variables in Σ appearing at the leaves. Where V is understood, we will often write Trees(Σ) for this set. We will often use the set X of canonical variables x1,x2,... . We denote Xn the set x1,..,xn. We define the set of terms over V to be the direct sum of the sets Trees(Xn; V) for each n. The significance of this perhaps roundabout definition is that every term can be uniquely identified as k-ary for some k. This allows us to treat any term as a function on trees:

If t is a n-ary term, and t1,..,tn are trees, define t(t1,..,tn) to be the tree

t:(lambda(xi)(ti))

where this denotes the expected substitution operation. Note that the lambda expression is only defined for xi in Xn.

This merely extends the normal notation s(t1,..,tn) for s in Vn. Thus individual letters in V and terms over V may be considered as functions on trees. We confirm that the set of terms is in fact a clone, and is the clone generated by V. We denote this clone V*. This is the first of the two pretheories.

The second pretheory is somewhat more involved. It is the clone of the theory which Thatcher calls the theory of finite state transformations on trees. I will not describe it fully here, but its n-ary symbols are maps from S (the state set) to Terms(Xn x S, V), and the function associated with such a map t is

lambda(t1,..,tn)(lambda(s)(t(s):(lambda(xi,s)(ti(s)))))

This is seen to mimic the linking action of a finite-state transformation. Note also that the carrier of the pretheory is the set of maps from S to variable-free trees over V. We confirm that this algebra is a pretheory, and we denote it V*S.

We recall that a finite-state transformation on trees (or FST) is a certain type of function from Trees x S to Trees. Note also that a Cl-morphism from V* to V*S takes a tree and yields a map from S to trees. Thus an expression f(t)(s) is a tree if f is such a morphism. We are now in a position to state the main theorem of this section:

Theorem. (i) If f is an FST, it induces a Cl-morphism f* from V* to V*S by

f*(t) = lambda(s)(f(t,s))

(ii) Every Cl-morphism f from V* to V*S induces a map f↓ from Trees x S to Trees by

f↓(t,s) = f(t)(s)

and f↓ is an fst.

(iii) f↓* = f; f*↓ = f.

In the course of the development, we also prove that all FSTs are total. The key to that argument is that certain functions are defined for every argument with which they are called. It turns out that the requirement that function symbols be uniquely defined as to arity, which heretofore seemed a rather technical point, and the requirement that Cl-morphisms map n-ary functions into n-ary functions are a model of this variable-binding argument. Furthermore, it develops that the requirement that Cl-morphisms preserve the projections is a mirror of Thatcher's 'boundary condition' in his definition of FST. These observations, as well as the actual result, give us a greater insight into finite-state transformations.

1. V-algebras and clones.

In this section we define V-algebras and an operation (closure) which generalizes the operation of taking the semigroup of a finite state machine.

Definition. A ranked alphabet is a pair $\langle V, r\rangle$, where V is a set and r is a function from V to the nonnegative integers. Where no confusion results, we refer to the ranked alphabet $\langle V,r\rangle$ by V alone. We denote by V_n the set $\{s \in V ; r(s) = n\}$. We will often call this set Vn.

A ranked alphabet is nothing more than a collection of function symbols (names) with indicated functionalities given by the function r. We interpret such an alphabet by specifying a function on an appropriate domain for each fucntion name. The resulting structure is called a V-algebra, or just an algebra.

Definition. If V is a ranked alphabet, a V-algebra $\mathcal{A}$ is a pair $\langle A, d\rangle$ where A is a set and $d:V \dashrightarrow [A^* \to A]$ such that if $s \in Vn$ then $d(s):A^n \dashrightarrow A$. We say A is the carrier of $\mathcal{A}$. We sometimes write $s^{\mathcal{A}}$ for d(s) and write $\langle A, V\rangle$ for $\langle A, d\rangle$, purposely confusing the alphabet V of function symbols and the set d[V] of functions. We will use capital script letters $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ to denote algebras, and roman letters A, B, C to denote their carriers.

Let us consider very briefly a few examples of algebras.

Example 1. Let G be a group. Let P = [.], with r(.) = 2. (so $\langle P,r\rangle$ is a ranked alphabet). Let i(.)(g,h) = gh (multiplication in G). Then $\langle G,i\rangle$ is a P-algebra.

Example 2. Let M be a monoid of operations on a set A. Make M a ranked alphabet by setting r(m) = 1 for each m in M. Let $d(m) :A \to A$ be given by d(m)(a) = m(a). Then $\langle A, d\rangle$ is a M-algebra.

Example 3. Let V be a finite ranked alphabet. Let Ξ be any set. Let Trees(Ξ; V) denote the set of trees built up from V with variables in Ξ as follows:

i) if $z \in \Xi$, then $z \in$ Trees(Ξ)

ii) if $s \in V0$, then $s \in$ Trees.

iii) if $s \in Vn$, and t1,...,tn $\in$ Trees, then the tree

```
          s
      /  /  \
    t1  t2 ... tn
```

is in Trees. We denote this tree s(t1,...,tn).

We have made here a couple of changes from normal mathematical notation. First, we will allow ourselves to use syntactic variables with names longer than a single letter (e.g. Trees, t1). We will use the symbol "." to denote multiplication, concatenation, or other normally notation-free operations where this may cause confusion. We will occasionally write things like lambda(i)(...xi...). It is hoped that this usage is at least moderately clear. Secondly, note the order of variables in Trees(Σ; V). This will allow us to drop later arguments when they are clear from context, e.g. Trees(Σ, V) may be abbreviated Trees(Σ), or just Trees, as, indeed, it is in the definition. If Σ is empty we write Trees (V) for Trees (Σ;V).

Let i be defined as follows:

i(s) = s if $s \in V0 \cup \Sigma$

i(s)(t1,...,tn) = s(t1,...,tn) if s is n-ary.

Then <Trees(V), i>is a V-algebra. It is called the generic or totally free V-algebra, since it can be shown that any V algebra is a homomorphic image of the generic V algebra.

Example 2, the monoid, has an interesting property. Let m,n,o,p,... be elements of M (ie functions A -> A) We can compose these functions to get new functions, ie lambda(x)[m(n(p(x)))]. M has the property that, whatever composition one creates, there is always a single function in M which gives the same map. This property, closure under composition, is extremely important. We generalize it to functions of more than one variable as follows:

Definition. (P. Hall) Let V be a set of operations on some domain, and let Vn denote the set of n-ary operators in V. We say V is a closed set of operations (or clone, for short) iff

1) for each n > 0, the function e_i^n = lambda(x1,...,xn)(xi) $\in$ Vn and

2)if g is n-ary, and f1,...,fn are k-ary operations in V, then the operation

lambda(x1,...,xk)(g(f1(x1,...,xk),...,fn(x1,...,xk)))

is in Vk. We denote this operation g(f1,...,fn).

Example 4. Let R be a ring. Let P be the set of polynmials with coefficients in R and variables in X = [x1,.. xn,...]. Then <R, P> is a P-algebra. (Notice that we have used the 'natural' interpretation of polynomials as functions on R). Furthermore P is a clone, since p(p1,...,pn) is just the polynomial obtained by substituting pi for xi uniformly in p. Note that R is just P0.

Definition. Let X =[x1,...,xn,...]. Let Xn = [x1,...,xn]. Note X0 is the empty set.

Note that the definiton of the set Trees is very much like the definition of the set of terms over a set of function symbols and constants. This duality is crucial to many of our arguments. We therefore define:

Definition: Let V be a ranked alphabet. Define the set of n-ary terms over V to be the set Trees(Xn; V). We denote this set Terms(n; V). As usual, we will delete the second argument where it is clear from context.

Definition. The set of terms over V, denoted Terms(V) is defined as the direct sum of the sets Terms(n; V) for every non-negative n.

Note that the sets Terms(n) are non-disjoint; in fact Terms(n) $\subset$ Terms(n+1). In the set Terms, however, we can uniquely identify any term as to its 'arity' (by the definition of direct sum). Thus in our definitions we will say 'let t be a term. If t is k-ary....' and this will be non-ambiguous.

We can now make Terms(V) a ranked alphabet by letting Terms_n be the image of Terms(n). To each t in Terms_n, we assign a function i'(t) as follows: Given t1,...,tn, i'(t)(t1,...,tn) is the tree resulting from substituting ti for xi in t. This substitution is to be done simultaneously for all i and for all occurrences of each xi. (This idea will be made somewhat more rigorous later. See Thatcher[1969]).

Example 5. <Trees(V), i'> is a Terms(V)-algebra. Furthermore the set i'[Terms] forms a clone. In fact, the desired function g(f1,...,fn) is obtained by applying i'(g) to the terms (trees) f1,...,fn. The reader will verify that a) this operation is well-defined, even though f1,...,fn are not variable-free, and b) that it gives the desired function. (Hint: prove that substitution is associative). The reader will note also that the n-ary term xi yields the i-th n-ary projection function e_i^n.

This example is a key one; we will refer to it many times in the future.

Note that we can naturally extend every operation in a clone V to operate on arguments in V via:

Let f1,..,fn $\in$ Vk, g $\in$ Vn. Then let g(f1,..,fn) be the composition of g with the f's as in the definition of clone. This composite is in Vk since V is a clone. So for each k, g:$(Vk)^n$ -> Vk. So for any k we have an algebra <Vk,V>.

Definition. An algebra <V0, V>, where V is a clone, is called a pretheory.

Example 6. <Trees(V),Terms(V)> is a pretheory. This pretheory is called the generic pretheory over V, and is denoted $\overline{V}$.

Definition. Let $\mathfrak{A}$ = <A, d> be a V-algebra. Let $\overline{V}$ be the smallest clone containing d[V]. Then $\overline{\mathfrak{A}}$, the closure of $\mathfrak{A}$, is defined as <A,$\overline{V}$ >.

$\overline{V}$, as defined above, is just the closure of the algebra <Trees(V), V> (The algebra of Example 3).

Proposition. For any algebra $\mathfrak{A}$, $\overline{\overline{\mathfrak{A}}} \approx \overline{\mathfrak{A}}$.

Proof. Trivial from the definition.

Our next goal is to provide an explicit construction of $\overline{\mathfrak{A}}$. This will, hopefully, make the concept of clone quite a bit clearer.

Definition. Let $\mathfrak{A}$ = <A, d> be a V-algebra. We define P($\mathfrak{A}$), the polynomial algebra over $\mathfrak{A}$, as follows: P($\mathfrak{A}$) = < A, d'> will be a Terms(V)-algebra. d' is defined as follows:

i) if s $\in$ V, then d'(s) = d(s)

ii) if xi $\in$ $Terms_n$, then d'(xi) = e_i^n

iii) Let t $\in$ $Terms_n$. Then t = s(t1,..,tk) for some t1,..tk in $Terms_n$ and s in V_k. Then let d'(t) = lambda(x1,..,xn)[d(s)(d'(t1)(x1,..,xn),..,d'(tk)(x1,..,xn))].

Note that d'[Terms(V)] consists of all the functions A^n --> A obtained by composing functions in $\mathfrak{A}$ (ie by composing thefunctions in d[V]). Thus going from $\mathfrak{A}$ to P($\mathfrak{A}$) is analogous to taking the semigroup of a finite-state machine, since the latter

operation is performed by taking all the functions Q --> Q obtainable by concatenating elements of the input alphabet V (regarding the elements of V as maps Q --> Q).

Theorem. $d'[\text{Terms}] = \overline{V}$

Proof: We need to show that $d'[\text{Terms}]$ is a clone, and that any clone containing V contains $d'[\text{Terms}]$.

To see that $d'[\text{Terms}]$ is a clone, let $d\leftarrow$ be a right inverse of d' (ie $d\leftarrow(f)$ is some tree such that $d'(d\leftarrow(f)) = f$). One may then easily confirm that $f(g1,\ldots,gn) = d'(i'(d\leftarrow(f))(d\leftarrow(g1),\ldots,d\leftarrow(gn)))$ (i' is the tree-substitution function of example 5).

Now let C be some clone containing V. We want to show that $d'[\text{Terms}] \subset C$. We do this by induction on the depth of $t \in$ Terms.

If $t \in V0$, then $d'(t) \in C$.
If $xn \in \text{Terms}_n$, then $d'(t)$ is $e_i^n \in C$
Otherwise t is $s(t1,\ldots,tk)$. By the induction hypothesis, $d'(ti) \in C$ for each i. But then $d'(t) = d'(s(t1,\ldots,tk)) = \text{lambda}(x1,\ldots,xn)[d'(s)(d'(t1)(x1,\ldots,xn),\ldots,d'(tk)(x1,\ldots,xn))] = d'(s)(d'(t1),\ldots,d'(tk))$ in C, so $d'(t)$ is in C. QED.

Thus we have the closure operation, as well, as analogous to taking the semigroup of an automaton. In the remainder of this paper we will discuss clones, occasionally referring to this characterization theorem where needed.

As usual, we will define a homomorphism for clones.

Definition. Let V, W be clones. A Cl-morphism (clone homomorphism) is a map F:V --> W such that

i) $F[Vn] \subset Wn$

ii) $F(e_i^n) = e_i^n$

iii) $F(f(g1,\ldots,gn)) = (F(f))(F(g1),\ldots,F(gn))$

Since any clone uniquely defines a pretheory (and viceversa), we say that a homomorphism between pretheories is just a Cl-morphism on the operators, with the map on the carriers regarded as the restriction of the Cl-morphism to V0.

Proposition. Cl-morphisms compose.

Proof. Trivial.

2. Theories and Pretheories

In this section we will explore the relationship between Lawvere's notion of an algebraic theory or algebraic category and our notion of pretheory. In order to make Lawvere's ideas clearer, we will introduce an intermediate notion, which we will call a 'complete algebra.'

Definition. A complete algebra is a pair

$\mathcal{A}$ = <A, {A(m,n) ; m,n integers}>

where A(m,n) is a set of maps A^m --> A^n such that

CA1) If f $\in$ A(m,n) and g $\in$ A(n,p), then their composition, g.f $\in$ A(m,p).

CA2) If f:{1,..,n} --> {1,..,m}, there exists a function f← in A(m,n) such that f←(x_1,...,x_m) = ($x_{f(1)}$,...,$x_{f(n)}$). Thse maps are called the trivial maps.

CA3) If f1,..fn $\in$ A(m,1),then there is a unique g $\in$ A(m,n) such that

g(a1,..,am) = (f1(a1,..,am),...,fn(a1,..,am))

Note that the value of g(a1,..,am) is indeed an n-tuple of elements of A. We write <f1,..,fn> for this unique g.

A complete algebra is not an algebra; it is the generalization of 'algebra' to functions into tuples of elements of A. We now proceed to characterize the complete algebras.

Lemma. If $\mathcal{A}$ is a complete algebra, then the direct sum of the sets A(n,1) forms a clone.

Proof. If f:{1} --> {1,..,n} := lambda(x)[i],then f← = e_i^n.

f(g1,..,gn) = f.<g1,..,gn>. QED.

Definition. If $\mathcal{A}$ is a complete algebra, we call the algebra <A, $\bigcup${A(n,1)}> the base of $\mathcal{A}$ and denote it $\mathcal{A}$*.

Definition. An algebra <A, V> is said to be closed iff V is a clone.

Lemma. Let $\mathcal{A}$= <A, V> be a closed algebra. Let

A(m,n) =
{lambda(x1,..,xm)[(f1(x1,..,xm),..,fn(x1,..,xm))] ; f1,..,fn$\in$

Vm}.

Then <A, [A(m,n)]> is a complete algebra. In fact, it is the smallest complete algebra containing $\mathcal{A}$, in the sense that if <A, B(m,n)]> is another complete algebra such that Vn $\subset$ B(n,1), then A(n,m) $\subset$ B(n,m) for all n,m.

Proof: We must verify that the sets A(m,n) satsfy conditions CA1 - CA3

1) Let f $\in$ A(m,n) , and g $\in$ A(n,p). Then f = (f1,..,fn) for some set of fi $\in$ Vm, where the tuple notation indicates taking the combining process of the hypothesis. (Note that this decomposition is well-defined). Similarly, g = (g1,..,gp) (gi $\in$ Vn). Let a denote an m-tuple (a1,..,am) of elements of A. Then

g.f(a) = g(f1(a),..fn(a))

=(g1(f1(a),..,fn(a)),..,gp(f1(a),..,fn(a))).

But since V is a clone, for each gi,

lambda(x1,..,xn)[gi(f1(x1,..,xn),..,fn(x1,..,xm))] $\in$ Vm

Call this function hi. So g.f =(h1,..,hp) $\in$ A(m,p).

2) Let f:{1,..,n} --> {1,..,m}. Then f← = $(e^{m}_{f(1)},\ldots,e^{m}_{f(m)})$.

3) This is guaranteed trivially by the construction.

Note also that the 'smallest' condition follows directly from condition 3, for if f $\in$ A(m,n), then f = (f1,..,fn) for some set of fi's in Vm (and therefore in B(m,1)). So f is in B(m,n) by CA3. QED.

We call this algebra $\langle\mathcal{A}\rangle$, the completion of $\mathcal{A}$.

Corollaries. If $\mathcal{A}$ is closed, then $\langle\mathcal{A}\rangle^{*} \approx \mathcal{A}$. If $\mathcal{A}$ is complete, $\langle\mathcal{A}^{*}\rangle \approx \mathcal{A}$.

Proof. Easy. We go from closed algebras to complete algebras by taking n-tuples; we go in the other direction by simply restricting ourselves to the 1-tuples.

Having established the intimate connection between closed algebras and complete algebras, we now proceed to the second half of our exposition: the connection between complete algebras and algebraic theories. For completeness, we begin

with a standard definition.

Definition. A category T consists of

i) a set Obj(T) of objects of T

ii) for each $A1$, $A2 \in$ Obj(T) a set $T(A1,A2)$ called the set of T-morphisms:$A1 \dashrightarrow A2$

iii) For each $A1,A2,A3 \in$ Obj(T) a map '.'

$.:T(A2,A3) \times T(A1,A2) \dashrightarrow T(A1,A3)$

satisfying the following axioms:

C1) If $f \in T(A1,A2)$, $g \in T(A2,A3)$, and $h \in T(A3,A4)$, then $h.(g.f) = (h.g).f$

C2) For each $A \in$ Obj(T) there exists a T-morphism $1_A \in T(A,A)$ such that for any $f \in T(A,B)$, $f.1_A = f = 1_B.f$

We often write $f:A \dashrightarrow B$ for $f \in T(A,B)$. Note that we are using 'left notation': $g.f$ means roughly 'f then g'.

The study of categories can be regarded as the study of generalized composition. Thus if the objects are sets and the morphisms functions between them, then the resulting object is a category. However, the morphism need not be set-theorete functions $A \dashrightarrow B$. In fact, one can choose $T(A,B)$ to be the set of all functions $B \dashrightarrow A$ (!). The reader may verify by appropriate symbol-pushing that the resulting object is a category.

Notation. Let [n] denote the set $\{1,..,n\}$. Note that [0] is the empty set, and [1] is a singleton. We will often write I for [1], and 0 for [0].

Notation. Let i_n denote the map $I \dashrightarrow [n]$ whose graph is $\{(1,i)\}$. Do not confuse i_n, the map $\{(1,1)\}$ in $T(I,[n])$, with $1_{[n]}$, the identity on [n].

Definition. (Lawvere [1963], Eilenberg and Wright [1967]) A theory T is a category such that

T1) Obj(T) = $\{[n] ; n = 0,1,2,3,...\}$

T2) $S0$, the category of sets [n] with all the set-theoretic functions between them as morphisms, is a subcategory of T (ie all the $S0$-morphisms are T-morphisms, and the identities and composition rules agree).

T3) Given any set of T-morphisms

$f1, \ldots, fn : I \dashrightarrow [p]$

there exists a unique T-morphism $g : [n] \dashrightarrow [p]$ such that

$$
\begin{tikzcd}
I \arrow[r, dashed, "i_n"] \arrow[dr, "fi"'] & {[n]} \arrow[d, "g"] \\
& {[p]}
\end{tikzcd}
$$

commutes for each $i \leq n$. We write $\langle f1, \ldots, fn \rangle$ for this unique morphism.

Thus for any $f : [n] \dashrightarrow [p]$, $f = \langle f.1_n, \ldots, f.n_n \rangle$ A more useful form of this statement is that any $f \in T([n],[p])$ is expressible as $\langle f1, \ldots, fn \rangle$ for some unique set of fi in $T(I,[p])$.

The notion of an algebraic theory is admittedly an obscure one. The primary purpose of this section is to show how this notion can be reduced to the simpler notion of an algebra (actually pretheory). We will now show the relationship between theories and complete algebras. To do this, we will first show how to construct a theory given a complete algebra. This construction will hopefully shed some light on the source of the definition of a theory.

Definition. Let $\mathcal{A}$ be a complete algebra $\langle A, \{A(m,n)\} \rangle$. Let $Th(\mathcal{A})$ be the category T constructed as follows:

1) $Obj(T) = \{[n]\}$

2) $T([m],[n]) = A(n,m)$ (note the reversal.)

3) Composition in T is the reverse of composition in $\mathcal{A}$ i.e., if $f \in T([n],[m])$ and $g \in T([m],[p])$, then g.f in T is just f.g in $\mathcal{A}$. The reader should confirm that this makes sense.

We call $Th(\mathcal{A})$ the theory of $\mathcal{A}$. (See Eilenberg and Wright [1967] for an alternative construction). We must, of course, confirm that what we have constructed is a theory. While the proof is trivial symbol-pushing, the astute reader will note the parallel between the axioms of a complete algebra and the axioms for a theory. So:

Proposition. If $\mathfrak{A}$ is a complete algebra, then Th($\mathfrak{A}$) is an algebraic theory.

Proof. 1. Th($\mathfrak{A}$) is a category. Closure under composition follows directly from the closure of complete algebras under composition(CA1). Associativity (C1) therefore follows from the associativity for set-theoretic functions. From CA2, we know that the identity on A^n is in A(n,n) for each n (it is the function induced by the identity of [n]). Therefore it serves as the identity $1_{[n]}$ in T([n],[n]).

T1. Obj (T) = {[n]}.

T2. S0 is a subcategory of T via the insertion sending f:[n] --> [m] to f*:A^m --> A^n. Note that f $\in$ S0([n],[m]), so f* $\in$ A(m,n), so f* $\in$ T([n],[m]), as desired. Note that (f.g)* = f*.g*, and ($1_{[n]}$)* (ie * of the identity on [n]) is 1_{A^n} (the identity on A^n). From these two observations it is easy to confirm that composition and identities agree on S0 (or its image) and on T, so S0 is indeed a subcategory of T. Note that this fact depends almost entirely on the axiom CA2, which guarantees the existence of all the trivial maps in $\mathfrak{A}$. Thus the axiom T2 requiring S0 as a subcategory may be viewed as simply requiring all the trivial maps to be in T.

T3. Given any set of T-morphisms f1,..fn: I --> [p], we want to find the unique T-morphism <f1,..,fn>. f1,..,fn are just maps in A(p,1), so let g = <f1,..,fn> in A(p,n) (whose existence is guaranteed by CA3). We would like g.I_n = fi in T. g.I_n in T = (I_n)*.g in $\mathfrak{A}$ = e_i^n.<f1,..,fn> = fi in $\mathfrak{A}$ = fi in T. So the axiom T3 may be viewed as requring the existence of direct products (CA3). QED.

Definition. If $\mathfrak{A}$ is an algebra (rather than a complete algebra), then let Th($\mathfrak{A}$) denote Th(<$\mathfrak{A}$>).

Having established that to every algebra there corresponds a theory, we naturally ask the next question: is every theory the theory of some complete algebra? To answer this question we will associate a complete algebra with each theory.

Definition. The free clone over T, Cl(T), is the set of maps T(I,[0])^n --> T(I,[0]) defined as follows: To each T-morphism f $\in$ T(I,[n]) we associate the map in Cl(T)$_n$ given by

lambda(a1,..,an)[<a1,..,an>.f]

Note that this definition makes sense, for , if a1,..,an $\in$ T(I,0), then <a1,..,an> $\in$ T([n],0), and f $\in$ T(I,[n]), so the

composition <a1,..,an>.f is defined and is a member of T(1,0). We leave it as a simple exercise to show that Cl(T) is really a clone.

Definition. The free complete algebra over T, denoted CA(T), is <Cl(T)>, the completion of the clone of T (or actually the completion of the pretheory <Cl(T)$_0$, Cl(T)>.

To get the desired result we will develop a series of technical lemmas.

Lemma. There is a natural bijection i:(T(1,[n]))^m --> T([m],[n]).

Proof. i is given by (f1,..,fm) --> <f1,..,fm>. 1-1-ness and onto-ness are both trivial.

Corollary. A theory T is uniquely determined by the morphisms T(1,[n]).

This is analogous to the statement that a complete algebra is uniquely determined by its base.

Theorem. T1 ≅ T2 iff CA(T1) ≅ CA(T2).

Proof. --> trivial. <--: It will suffice to show Cl(T1) ≅ Cl(T2) --> T1 ≅ T2. But it is clear that Cl(T1) ≅ Cl(T2) --> T1(1,[n]) ≅ T2(1, [n]). But then, by the last corollary, T1 ≅ T2. QED.

Definition. A complete algebra is said to be self generated iff A is isomorphic to A(0,1).

Corollary. A complete algebra is self generated iff it is the completion of a pretheory.

Theorem. If $\mathfrak{A}$ is a pretheory, then $\mathfrak{A} \cong$ Cl(Th($\mathfrak{A}$))

Proof. Let $\mathfrak{A}$ = <V0, V>. Let T denote Th($\mathfrak{A}$). Let $\mathfrak{B}$ denote Cl(T). We know that T(1,[n]) is just Vn, and that $\mathfrak{B}$n is just T(1,[n]). So we have a natural bijection e:Vn --> $\mathfrak{B}$n for each n. It remains only to show that the map e is a homomorphism, ie

e(f(a1,..,an)) = e(f)(e(a1),..,e(an)).

Now e is the identity on the 0-ary elements, so we need only show that e(f) = f. Now the image of f in T is just the morphism f. The image of f in Cl(T) is the map

lambda(a1,..,an)[<a1,..,an>.f]

Now the . in that expression is the composition of Th($\mathcal{A}$), which is defined to be just the reverse of composition in <$\mathcal{A}$>. So <a1,..,an>.f in T is f.<a1,..,an> in <$\mathcal{A}$>, which is just f(a1,..,an). So the image of f in $\mathcal{B}$ = Cl(T) is

lambda(a1,..,an)[f(a1,..,an)] = f, as desired. QED.

Corollary. If $\mathcal{A}$ is a self generated complete algebra, then $\mathcal{A} \simeq$ CA(Th($\mathcal{A}$)).

Corollary. If T is a theory, then T $\simeq$ Th(Cl(T)).

Proof. Let $\mathcal{A}$ be Cl(T) in the theorem. Then Cl(T) $\simeq$ Cl(Th(Cl(T))). But by the first theorem in this section, this implies that T $\simeq$ Th(Cl(T)). QED.

What we have shown so far is that there is a natural correspondence

$$\begin{tikzcd} \text{theory} \arrow[d, leftrightarrow] \\ \text{complete algebra} \arrow[d, leftrightarrow] \\ \text{pretheory} \end{tikzcd}$$

such that all of the arrows commute. In fact, we can assert something strnger. If we define a homomorphism of theories (as we will very shortly), we will get the following ladder diagram to be commutative:

$$\begin{tikzcd} T1 \arrow[r] \arrow[d, leftrightarrow] & T2 \arrow[d, leftrightarrow] \\ CA1 \arrow[r] \arrow[d, leftrightarrow] & CA2 \arrow[d, leftrightarrow] \\ Cl1 \arrow[r] & Cl2 \end{tikzcd}$$

where the vertical arrows are any of the correspondences of the previous diagram, ad the horizontal arrows are the appropriate homomorphisms. In fact, specifying any one of the horizontal arrows will induce naturally the other two horizontal arrows.

Furthermore, all of the inductions commute, e.g. if a Cl-morphism f induces a Th-morhpism f*, then the Cl-morphism induced by f* will be precisely f.

Our first task is to define homomorphisms for theories.

Definition. A Th-morphism (theory homomorphism) T1 --> T2 is a functor preserving S0, ie a map F on the morphisms of T1 such that

TM1) F:T1([n],[m]) --> T2([n],[m])

TM2) if f $\in$ T([n],[m]) and g $\in$ T1([p],[n]) then

F(f.g) = Ff.Fg

TM3) if f $\in$ S0, then Ff = f .

One should not confuse a Th-morphism, which is a map between two theories, with a T-morphism (or, say, Th($\mathfrak{A}$)-morphism), which is a map inside a particular theory. We will avoid naming any theory Th, so this notation will be unambiguous.

Definition. A CA-morphism (Complete algebra homomorphism) from $\mathfrak{A}$ to $\mathfrak{B}$ is a map F whose domain is A $\cup$ $\bigcup${A(m,n)} such that

1) F(A) $\subset$ B

2) F:A(m,n) --> B(m,n)

3) F(f.g) = Ff.Fg

4) If f is a trivial map, Ff = f.

Both Th- and CA- morphisms may be viewed as homomorphisms (conditions 3 and TM2) which preserve the dimensionalities of their domains and ranges(conditions 2 and TM1), and which preserve the trivial maps (conditions 4 and TM3). Similarly, one can view Cl-morphisms in the same light. The ladder theorem, then, should not be greatly surprising.

We will also need the technical result that Th- and CA-morphisms preserve their direct product operations as well.

Proposition. Let F be a Th-morphism, f1,..,fn $\in$ T(1, [k]). Then F(<f1,...,fn>) = <Ff1,..,Ffn>

Proof. $F(\langle f1,..,fn\rangle).i = F(\langle f1,..,fn\rangle).F(i) = F(\langle f1,..,fn\rangle.i) = Ffi$, for each i. By the uniqueness condition of the definition of $\langle ... \rangle$, this is sufficient to show that $F(\langle f1,..,fn\rangle) = \langle Ff1,..,Ffn\rangle$. QED.

The proof for CA-morphisms is similar.

Theorem. i) Let $\mathcal{A},\mathcal{B}$ be complete algebras, $F:\mathcal{A} \dashrightarrow \mathcal{B}$ be a CA-morphism. Then the restriction of F to the sets $A(m,1)$, denoted $M1(F)$, is a Cl-morphism on the clones $\mathcal{A}* \dashrightarrow \mathcal{B}*$ satisfying the ladder condition of the previous discussion.

ii) Let $\mathcal{A},\mathcal{B}$ be clones, $F:\mathcal{A} \dashrightarrow \mathcal{B}$ a Cl-morphism. Then F induces a CA-morphism $M2(F):\langle\mathcal{A}\rangle \dashrightarrow \langle\mathcal{B}\rangle$ satisfying the ladder condition.

iii) $M1(M2(f)) = f$; $M2(M1(f)) = f$

Proof. i. Trivial.

ii) M2 is given by: $M2(F)$ on A is just F. Let $f \in A(m,n)$ be $\langle f1,..,fn\rangle$. Then let $M2(F)(f) = \langle Ff1,..,Ffn\rangle$. We need to confirm that $M2(F)$ is a CA-morphism. Conditions 1 and 2 hold trivially. To verify condition 3, let $M2(F)$ be denoted by G. Let $f = \langle f1,..,fn\rangle$, $g = \langle g1,..,gm\rangle$. Then

$$
\begin{aligned}
G(f.g) &= G(\langle f1.g,..,fn.g\rangle) = \langle F(f1.g),..,F(fn.g)\rangle \\
&= \langle F(f1(g1,..,gm)),..,F(fn(g1,..,gm))\rangle \\
&= \langle F(f1)(F(g1),..,F(gm)),..,F(fn)(F(g1),..,F(gm))\rangle \\
&= \langle F(f1).\langle F(g1),..,F(gm)\rangle,..,F(fn).\langle F(g1),..,F(gm)\rangle\rangle \\
&= \langle F(f1).G(g),..,F(fn).G(g)\rangle \\
&= \langle F(f1),..,F(fn)\rangle.G(g) \\
&= G(f).G(g)
\end{aligned}
$$

To verify that the trivial maps are preserved, note that the trivial maps are the direct products of the projections, which are preserved by F. Note that the ladder condition follows immediately from the natural inclusion of $\mathcal{A}$ in $\langle\mathcal{A}\rangle$.

iii) Obvious, since M2 on $A(n,1)$ is the identity.

Theorem. Let T1, T2 be theories, $F:T1 \dashrightarrow T2$ a Th-morphism. Then F induces a Cl-morphism $M3(F):Cl(T1) \dashrightarrow Cl(T2)$, such that the following diagram commutes:

$$\begin{tikzcd} T1 \arrow[r, "f"] \arrow[d, leftrightarrow] & T2 \arrow[d, leftrightarrow] \\ Cl(T1) \arrow[r, "M3(F)"'] & Cl(T2) \end{tikzcd}$$

where the vertical arrows are the natural insertions of the construction of Cl(T).

Proof. Let $s \in Cl(T1)_n$. Then s is the image of a unique morphism (also denoted s) in T1(1,[n]). Let M3(F)(s) be the image of F(s) in Cl(T2). $M3(f)(s) \in Cl(T2)_n$. M3(F)(s(s1,..,sn)) = F(<s1,..,sn>.s) = F(<s1,..,sn>).F(s) = <F(s1),..,F(sn)>.F(s) = M3(F)(s)(M3(F)(s1),..,M3(F)(sn)), so the homomorphism condition holds. The projections are in S0, so they are preserved. So M3(F) is a Cl-morphism. Again, the diagram condition is trivial, since the vertical arrows are insertions, and M3 is the identity on the inserted image.

Theorem. Let $\mathfrak{A}, \mathfrak{B}$ be complete algebras, F:$\mathfrak{A}$--> $\mathfrak{B}$ a CA-morphism. Then F induces a Th-morphism M4(F):Th($\mathfrak{A}$)-->Th($\mathfrak{B}$), such that the diagram condition holds.

Proof. We need only specify M4(F) on the morphisms of Th($\mathfrak{A}$). Denote M4(F) by G and Th($\mathfrak{A}$) by T. Let $f \in$ T([n],[m]). Then f is the image of some map (also denoted f) in A(m,n). Let G(f) be the image of the map F(f) in B(m,n). Again the diagram condition holds obviously. We must confirm that G a is Th-morphism. The dimensionality conditon, TM1, holds trivially from the corresponding dimensionality condition (#2) for CA-morphisms. S0 is preserved, since S0 is the image of the trivial maps in $\mathfrak{A}$, which are preserved by the CA-morphism. We need only confirm that the homomorphism condition (TM3) holds.

G(f.g) = F(g.f) = F(g).F(f) = G(f).G(g)

QED.

We have now established the following transformations between the various kinds of homomorphisms:

$$\begin{tikzcd}
\text{Th-morphisms} \arrow[dd, bend right=60, "M3"'] \\
\text{CA-morphisms} \arrow[u, "M4"'] \arrow[d, shift right, "M1"'] \\
\text{Cl-morphisms} \arrow[u, shift right, "M2"']
\end{tikzcd}$$

We have shown that M1 and M2 commute, so to complete our prgran, we need only one more result, which we will leave as an exercise to the reader.

Proposition. M4.M2.M3 = the identity; M3.M4.M2 = identity.

Proof. Trivial from the insertions.

3. Pretheories and Finite State transformations

In this section we will apply pretheories to the study of finite state transformations on trees. This investigation was motivated by a remark of Thatcher [1969]. He remarked that the finite state transformations on trees induced a homomorphism on a certain semigroup of substitutions, which he called a pretheory. He left open the question of an abstract definition of a pretheory, and conjectured that with the proper definition of pretheory every homomorphism of pretheories would yield a finite state transformation.

Definition. Let Z be some (infinite) set of variable symbols. Then we call any function $\eta : Z \dashrightarrow \mathrm{Trees}(Z,V)$ a substitution.

Definition. We define the substitution operator : as follows. Let $t \in \mathrm{Trees}(Z, V)$, η be a substitution. Then $t{:}\eta$ is defined by induction on the construction of t:

i) if $f \in Z$, then $t{:}\eta = \eta(t)$

ii) if $t = s(t1,\ldots,tn)$, then $t{:}\eta = s(t1{:}\eta,\ldots,tn{:}\eta)$, where $s \in Vn$, $n \geqslant 0$.

This definition is due to Thatcher, who also originated the following definition and proposition.

Definition. We can extend : to substitutions in the first argument as follows: if η, ξ are substitutions, then let

$$\eta : \xi = \mathrm{lambda}(z)(\eta(z) : \xi)$$

The reader will confirm that this definition makes sense.

Proposition (Thatcher) $(\eta{:}\xi){:}\zeta = \eta{:}(\xi{:}\zeta)$

Proof. See Thatcher 1969.

Corollary. The set $(\mathrm{Trees}(Z))^{Z}$ of substitutions on a set Z forms a monoid with the operation :.

Notation. For any $s \in Vn$, let s' denote the tree $s(x1,\ldots,xn) \in \mathrm{Trees}(Xn, V)$.

Proposition. Any tree in $\mathrm{Trees}(X, V)$ is decomposable as $s'{:}\eta$, where s' is uniquely determined and η is a substitution on X well defined on Xn.

Proof. This is just the definition of Trees(X) in slightly different notation. Let t be s(t1,..,tn) and let η be any substitution mapping xi to ti for xi in Xn. Then t = s':η.

Definition. Let S be a finite set. Let f0:V x S --> Trees(XxS, V) such that f0[V_n] $\subset$ Trees(Xn x S). We can extend f0 to f:Trees(X) x S --> Trees(X x S) by

i) t(xi,s) = ⟨xi, s⟩ ($\in$ X x S $\subset$ Trees(XxS, V))

ii) f(a':η, s) = f0(a,s):lambda(x,s)[f(η(x), s)]

We can regard the substitution in (ii) as the extension of f to substitutions:

f(η) = lambda(x,s)[f(η(x),s)]

Then f(a':η, s) = f0(a,s):f(η).

Any function f:Trees(X)xS --> Trees(XxS) defined in this way is called a finite-state transformation. The rationale behind this defintion is given in much more detail in Thatcher [1969]. The reader may also verify that this definition is equivalent to the more intuitive formlation of Rounds [1969]. One may regard the function f0 as specifying the set of productions, and the recursion schema as specifying the action of a gsm. A number of facts are easily proved about FSTs.

Proposition (i) f is total and well defined.
ii) f[Trees(V) x S] $\subset$ Trees(V)
iii) f(t:η,s) = t(t,s):f(η)

Proof. (i) and (ii) are proved via recursion induction. The key point that wants verifcation is the following: In evaluating f on a tree, we must evaluate f(η), for some η which is well defined ony on some initial segment Xn of X. We must confirm that f(η) is called only for arguments for wich it is defined. Similarly, we can show that if t is in Trees(V), then the boundary condition (i) of the defintion is never used (!) The proof is elementary and will not be reproduced here.

(iii) is proved by Thatcher [1969], q.v.

Proposition. f(η:ξ) = f(η):f(ξ)

Proof. Trivial from iii.

This was Thatcher's key observation: That FSTs induced homomorphisms on the semigroups of transformations (in

particular, in this case, the FST f induces a homomorphism (Trees(X))^{X} --> (Trees (XxS))$^{X\times S}$. He therefore called these monoids pretheories. Unfortunately, the converse was not true, and Thatcher offered the open problem of defining an algebraic structure in which the converse held. We will define two pretheories, such that a map is an FST if and only if it is a Cl-morphism between the two pretheories.

If one considers why the converse of the last proposition does not hold, one is led back to the detail of the proof that f is well-defined: It develops that one has to worry about the transformation being called with arguments for which it is not well-defined. One may offer necessary and sufficient conditions on a homomorphism for it to be the homomorphism of an FST. Roughly stated, the condition is that the homomorphism be bounded, in the sense that its value at x depend only on the value of the argument at x. This boundedness is again roughly reflected in the algebraic theory which Thatcher mentions as the theory of FSTs. It develops that any Th-morphism into that theory is an FST (in a natural sense). The 'boundedness' condition thus stated is merely a reflection of the conditions for a Th-morphism: that if F T([n],[m]), then Ff$\in$ F(T)([n],[m]), and that S0 is preserved. The pretheory we will offer is just the clone of the theory of Thatcher. This description is not meant to be clear, but merely suggestive of the considerations that guided this research. We will now present our finished product in a 'standard' mathematical presentation, ie the framework of this development will be (unfortunately) well-hidden.

Notation. Let V* denote $\bar{V}$, the free clone over V defined in Section 1. We recall that this is <Trees(V),Terms(V)>, with the interpretation that if t $\in$ Terms$_n$, and t1,..,tn are trees, then t(t1,..,tn) = t:lambda(xi)[ti]. (The reader may confirm that this is equivalent to the previous definition, with an appropriate reading of our somewhat informal lambda-notation). Note also that the substitution in question is well defined only on Xn, as desired.

Notation. Let Maps(S;A) denote the set of all functions f:S --> A.

Notation. Let the n-ary terms in XxS be the set Trees(Xn x S; V). Let the terms over XxS be the direct sum of the sets of n-ary terms over XxS. Denote this set by Terms (XxS). Make Terms(XxS) a ranked alphabet in the usual way by setting Terms(XxS)$_n$ equal to the n-ary terms.

Definition. If A is a ranked alphabet, make Maps(S;A) a ranked alphabet by setting r(f) = max [r(f(s)); s $\in$ S] (This

will always be finite since S is finite).

Definition. Let V be a ranked alphabet, S a finite set. Then let V*S denote the algebra

<Maps(S;Trees(V)), Maps(S;Terms(XxS; V))>

where t(t1,..,tn) = lambda(s)[t(s):lambda(xi,s)[ti(s)]]. We denote this operation t*(t1,..,tn) to distinguish it from the very-similar-looking, but quite different, operation in V*.

Let us try to explain this. The carrier of our algebra consists of maps from S into Trees. So an element of the carrier can be envisioned as an "S-tuple" (t1,..,ts) of variable-free trees over V. The operation symbols are S-tuples of trees which may have variables in X x S at their leaves. What are the operations of V*S? Let t1,..,tn be elements of V*S (that is, elements of the carrier, or maps from S to Trees), and let t be a n-ary operation symbol in V*S. Then we have to apply t to t1,..,tn and get another map from S to Trees. Envision t as an S-tuple (f1,..,fs) (remembering that these fi are trees with some variables in their leaves). Our operation yields an S-tuple (f1:¶,..,fs:¶) for an appropriate substitution ¶. The ¶ we have chosen says: If you are at a variable <xi,s> on fj, take argument ti (an S-tuple of nice, pure, variable-free trees), take its s-th element (one nice, pure, variable-free tree), and attach it to this leaf, thus turning fj into a N,P,V-F tree, and turning t into a map from S into NPV-F Trees, as desired.

This may look like so much legerdemain: after an appropriate flash of the notational magic wand, we have pulled an imaginary rabbit out of an imaginary hat. The remainder of this paper, however, will be devoted to demonstrating that real rabbits actually live in top hats or, more precisely, that V*S is a natural thing to study if one is intereseted in FSTs on trees.

Proposition. V*S is a pretheory.

Proof. The projection function e_i^n is provided by the n-ary term lambda(s)[<xi,s>], for

lambda(s)[<xi,s>]*(t1,..,tn) =

= lambda(s) [lambda(s) [<xi.s>](s):lambda(xi,s) [ti(s)]]

= lambda(s)[<xi,s>*lambda(xi,s)[ti(s)]]

= lambda(s)[lambda(xi,s)[ti(s)](xi,s)]

= lambda(s)[ti(s)]

= ti

To get closure, let t $\in$ Maps(S; Terms(XxS))$_n$, t1,..,tn $\in$ Maps(S; Terms)$_\ell$, u1,..,uk $\in$ Maps(S; Trees).

We want h $\in$ Maps S; Terms)$_k$ such that

h*(u1,..,uk) = t*(t1*(u1,..,uk),..,tn*(u1,..,uk))

Let u denote the k-tuple u1,..uk. Let h = t*(t1,..,tn). (Note that while * was not formally defined on non variable-free trees, there is nothing in the definition to prevent us from so using it. So we will).

(t*(t1,..,tn))*u = (lambda(s)[t(s)*lambda(xi,s)[ti(s)]])*u

= lambda (s) [[t(s):lambda (xi,s) [ti(s)]]:lambda (xj,s) [uj(s)]]

= lambda (s) [t(s):[lambda (xi,s) [ti(s)]:lambda (xj,s) [uj(s)]]

= lambda (s) [t(s):lambda (xi,s) [ti(s):lambda (xj,s) [uj(s)]]]

= lambda (s) [t(s):lambda (xi,s) [(ti*(u))(s)]]

= t*(t1*u,..,tn*u)

So V*S is closed. Note Trees $\subset$ Terms(XxS), so the carrier is just the 0-ary operators. QED.

Theorem. Every FST f induces a Cl-morphism

f*:V* --> V*S

by f*(t) = lambda (s) [f(t,s)].

Proof. We need only confirm that this is a Cl-morphism.

i) f*(e_i^n) = lambda (s) [f(xi,s)] = lambda (s) [<xi,s>], which is, as seen previously, the projection operator in V*S.

ii) f*(t(t1,..,tn)) =

= lambda (s) [f(t:lambda (xi) [ti],s)]

= lambda (s) [f(t,s):f(lambda (xi) [ti])]

= lambda (s) [f(t,s):lambda (xi,s) [f(lambda (xi) [ti](xi),s)]]

= lambda (s) [f(t,s):lambda (xi,s) [f(ti,s)]]

= lambda (s) [f*(t)(s):lambda (xi,s) [f*(ti)(s)]]

= (f*(t))*(f*(t1),...,f*(tn))

Q.E.D.

Although the 'linking mechanism' in V*S is admittedly obscure, this theorem shows how it is naturally related to the linking mechanism of the FST.

Theorem. Every CI-morphism f:V* --> V*S induces an FST (with state set S) via

f↓ = lambda (t,s) [F(t)(s)]

Proof. Note that for t ∈ Trees, f(t) ∈ Maps(S;Trees), so f(t)(s) ∈ Trees, as desired. Our proof will be by recursion induction. We will show that f↓ satsfies the recursion scheme in the definition of FST. Since we know that scheme is total, we conclude the f↓ is precisely given by it.

f↓(xi,s) = f(xi)(s) = e_i(s) = lambda (s) [<xi,s>](s) = <xi,s>

f↓(a(t1,...,tn),s) = f(a(t1,..tn))(s)

= (f(a)*(f(t1),...,f(tn)))(s)

= (f(a)(s)):lambda (xi,s) [f(ti)(s)]

= f↓(a,s):lambda (xi,s) [f↓(ti,s)]

QED.

Proposition. f*↓ = f; f↓* = f.

Proof. Trivial.

This completes the proof of our desired result: That the FSTs correspond, in a natural way, to the CI-morphsims

between the given pretheories.

BIBLIOGRAPHY

Lawvere [1963] Functorial Semantics of algebraic Theories.
Proc Natl Acad Sci USA 50:869-872.

Lawvere[1965] Algebraic theories, Algebraic Categories, and Algebraic Functors.
The Theory of Models, Proc. 1963 Intl Symp at Berkeley North-Holland, Amsterdam, 1965, pp 413-418.

Arbib + Giveon [1968] Algebra Automata I: Parallel Programming as a Prolegomena to the Categorical Approach.
Info + Contr 12:331-345.

Thatcher + Wright [1968] Generalized Finite Automata Theory with an Application to a decision problem of second-order logic.
Math Syst. Theory 2 (1968) 57-81.

Eilenberg + Wright [1967] Automata in General Algebras.
Info + Contr. 11:4 (1967) 452-470

Maclane + Birkhoff [1967] Algebra. MacMillan, NY, 1967.

Giveon + Arbib [1968] Algebra Automata II: The Categorical Framework for Dynamic Analysis. Infor + Contr. 12:346-370.

Gratzer [1968] Universal Algebra. Van Nostrand, Princeton 1968.

Thatcher [1969] Transformations and Translations from the Point of view of Generalized Finite Automata Theory
Proc. ACM Symp Th Comp, Marina Del Rey 1969, pp 129-142.

Rounds [1969] Context-Free Grammars on Trees
same Proc. p. 143.